NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

A CONTRASTING LOOK AT NETWORK FORMATION MODELS AND THEIR APPLICATION TO THE MINIMUM SPANNING TREE

by

Deanne B. McPherson

September 2009

Thesis Advisor:	David L. Alderson
Second Reader:	Timothy H. Chung

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2009	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE A Contrasting Look at Network Formation Models and Their Application to the Minimum Spanning Tree	**5. FUNDING NUMBERS**
6. AUTHOR(S) Deanne B. McPherson	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Networks are prevalent in man-made and natural systems throughout the world. Despite recent efforts to characterize and catalog networks of all kinds, there is considerably less known about the forces that drive network formation. For many complex systems, it is unclear whether networks are the result of an explicit effort to achieve some overarching global system objective, or if network structure is just a byproduct of local, selfish decisions. In this thesis, we review network formation models and conduct numerical experiments to contrast their behavior and the structural features of the networks they generate. We focus primarily on problems related to the formation of minimum spanning trees and consider the cost of selfish behavior, more commonly known as the price of anarchy, in network formation. We also explore differences between local, decentralized methods for network formation and their global, centralized counterparts.

.

14. SUBJECT TERMS Network formation, graph generation, minimum spanning tree, price of anarchy	**15. NUMBER OF PAGES** 67
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

A CONTRASTING LOOK AT NETWORK FORMATION MODELS AND THEIR APPLICATION TO THE MINIMUM SPANNING TREE

Deanne B. McPherson
Lieutenant, United States Navy
B.S., University of Scranton, 2001

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 2009

Author: Deanne B. McPherson

Approved by: David L. Alderson
Thesis Advisor

Timothy H. Chung
Second Reader

Robert F. Dell
Chairman, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Networks are prevalent in man-made and natural systems throughout the world. Despite recent efforts to characterize and catalog networks of all kinds, considerably less is known about the forces that drive network formation. For many complex systems, it is unclear whether networks are the result of an explicit effort to achieve some overarching global system objective, or if network structure is just a byproduct of local, selfish decisions. In this thesis, we review network formation models and conduct numerical experiments to contrast their behavior and the structural features of the networks they generate. We focus primarily on problems related to the formation of minimum spanning trees and consider the cost of selfish behavior, more commonly known as the price of anarchy, in network formation. We also explore differences between local, decentralized methods for network formation and their global, centralized counterparts.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AQM	Active queue management
BCG	Bilateral connection game
FKP	Fabrikant, Koutsoupias and Papadimitriou
HFN	Hastily formed networks
HOT	Highly optimized tolerance
IID	Independent identically distributed
MST	Minimum spanning tree
TCP	Transmission control protocol
UCG	Unilateral connection game
WWW	World-wide web

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Networks are prevalent in man-made and natural systems throughout the world. Despite recent efforts to characterize and catalog networks of all kinds, considerably less is known about the forces that drive network formation. For many complex systems it is unclear whether networks are the result of an explicit effort to achieve some overarching global system objective, or if network structure is just a byproduct of local, selfish decisions. We conduct numerical experiments using network formation models to examine the behavior and structural properties of the networks they form.

We discuss several models of network formation, including the Erdős and Rényi (1959) random graph model, the random geometric graph model, and a preferential attachment model, popularized by Barabási and Albert (1999), which produces a network with a node degree distribution that can be described by a power-law. We then review an optimization-based model, from which we derive our model for our numerical experiment as well as game theoretic models.

We provide a review of the minimum spanning tree (MST) problem. We introduce it as a formal optimization problem, which is non-trivial to solve as an integer linear program for large problems. We then review two centralized algorithms, Kruskal's (1956) and Prim's (1957), which take advantage of the special network structure in order to more easily solve the MST problem. In contrast to the global algorithms, we review the decentralized algorithm of Gallagher, Humbolt and Spira (1983) that utilizes "message passing" between nodes to solve for the MST.

Our numerical experiments use a simplified version of the optimization-based model, which grows networks by adding nodes to the unit square one at a time. Each new node forms an arc in the network to the node that minimizes the arc's Euclidian distance. By restricting the objective function to only distance, we determine the price of anarchy by comparing the total network cost to the optimal cost of the MST. The results, based on 10,000 trials, indicate the cost of a network formed with 100 nodes is

approximately 50% greater than the MST. By altering the arrival order of the nodes to the network, we observe that precedence plays an important role in these network growth heuristics.

We also consider network rewiring experiments, in which we allow the nodes to reevaluate their initial arcs to see if they can improve their objective function. They continue this process until no node can improve, and the network is in equilibrium. After 10,000 trials, the cost of this 100 node network is approximately 15% greater than the MST. We also alter the order in which the nodes reevaluate their arcs, and determine that precedence in the rewiring also affects the final network structure.

We conclude with suggestions for continued research to determine if there might be an interpretation of the local, myopic decision process we utilized that lends itself to an equivalent global solution. We appeal to the example of the Internet, in which duality theory has helped to understand the behavior of the complex network.

ACKNOWLEDGMENTS

I would like to acknowledge Professor David L. Alderson for all his patience, time, input, and guidance through the many revisions.

I would also like to acknowledge CPT Marco Draeger, who provided assistance and recommendations for coding the algorithms, and to CPT Jonathan Derosier who developed the program to generate the network pictures in this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A *graph* is a collection of *nodes* (also called *vertices*) connected by a set of *arcs* (also called *edges*). A *network* is a specific type of graph, where associated with each arc or node is additional information, such as the cost or capacity of the arc or the demand at a node. Networks are integral to a variety of systems that we rely upon each day. Our transportation system is made up of a variety of networks including road, rail and airline networks. Our electrical system is a network of wires that ensures power reaches homes and businesses. Communications systems, including the Internet, are expanding beyond the typical hard wired lines to include wireless networks. Even individuals' relationships with one another can be viewed as a network of social ties.

Each of these networks plays an important role in society. A transportation network provides a means for goods and people to move from a starting location to a destination. The electrical system continuously balances generation with fluctuating user demand. Communication networks and the Internet provide a massive increase in the amount of easily obtainable information, and they also dramatically decrease the amount of time required to transfer information around the world. The study of social networks is increasingly popular, with sites such as Facebook and Twitter capturing evolving relationships between millions of people. The analysis of networks is even helping to fight terrorism by identifying terrorist networks so that we can determine where it is most effective to disrupt them.

Because of the prevalence and importance of networks throughout our world, the last decade has seen increased scientific attention on the properties and functions of networks. Much of the recent effort has been to catalog a diversity of networks and to characterize their structural features. The majority of this work on network structure has emphasized the connectivity properties of the underlying graph, thus renewing interest in graph theory.

The study of networks is actually centuries old. Graph theory dates back to Leonard Euler in 1736 (Biggs, Lloyd and Wilson, 1998), when he proved there was no feasible solution to the Konigsberg Bridge Problem. The development of random graph theory in the 1940s and 1950s generated great interest in the characteristics of graphs and networks (see Newman, Barabási and Watts, 2006, for notable publications). Most recently, the advent of “network science” during the last decade has witnessed renewed interest in the large-scale properties of graphs (National Research Council, 2005).

Despite considerable effort devoted to the “what” of these networks, considerably less is known about the “how” and the “why.” It is not always clear what drives the formation of networks. A fundamental question is whether networks form to achieve some overarching global objective or if network structure is just a byproduct of local, selfish decisions. Or it may be a combination of the two. It is also not clear how global, centralized network formation versus local, decentralized formation affects the properties of the resulting network.

Understanding the forces that drive network formation is becoming increasingly important. Of particular interest is the Internet. This is an extremely complex network that has managed to evolve and grow at an amazing pace. To some researchers, the Internet exemplifies a system that has *self-organized*. They argue that the network was not built by a “central” designer, but arose rather as a result of the localized actions of the users and service providers. In spite of its ad-hoc construction, the Internet is still relatively robust (Willinger and Doyle, 2004). The ability to model such a complex network and to understand its underlying properties is extremely relevant for the study of both man-made and natural systems.

Another area of increasing importance is the use of Hastily Formed Networks (HFNs) in response to humanitarian aid and disaster relief operations, such as a Hurricane Katrina scenario (Denning, 2006). These types of networks require rapid coordination and information between a variety of agencies.

Research in the “how” of network formation has ranged from random graph generation to system design. Erdős and Rényi (1959) pioneered the exploration of random graphs models, which generated interest in graph and network theory. More recently, the study of network science has focused attention on “small-world networks” and “scale-free networks.”

Small-world networks (Watts and Strogatz, 1998) are networks that have high local clustering and have path lengths between arbitrarily chosen nodes that are still relatively short. Small-world networks have been used to explain the “six degrees of separation” phenomena.

Scale-free networks (Barabási and Albert, 1999) have been used to describe any number of complex, large real world networks whose nodes degree distributions tend to follow a power law. This observation of a power law can be seen in the World-Wide Web (WWW), biological sciences, and social networks.

In this thesis, we review some of the recent models used for network formation and conduct numerical experiments to compare and contrast the structural features of the graphs they generate. We construct all the algorithms we discuss as well as the numerical experiments in this thesis from scratch using the Java programming language. We focus primarily on problems related to the formation of *minimum spanning trees* and consider the cost of selfish behavior, more commonly known as the *price of anarchy*, in network formation. We then contrast some of the local, decentralized methods for network formation to the global, centralized methods. Our results help to clarify the tensions in network formation problems for both man-made and natural systems.

The remainder of this thesis proceeds as follows. Chapter II reviews previous research in network formation. Chapter III introduces the minimum spanning tree problem along with both centralized and decentralized algorithms for solving it. Chapter IV presents experiments that help to elucidate the underlying mechanism behind network formation in relation to the minimum spanning tree. We conclude in Chapter V with a brief summary of our findings, as well as provide guidance for future research in this area.

THIS PAGE INTENTIONALLY LEFT BLANK

II. NETWORK FORMATION MODELS

In this chapter, we review several models of network formation. Throughout this thesis, we adopt the following notation and definitions from Ahuja, Magnanti and Orlin (1993).

A *graph*, or *network*, $G = (N, A)$ consists of a set N nodes and a set of A arcs. The number of nodes is $n = |N|$ and the number of arcs is $m = |A|$. An arc from node i to node j is denoted as (i, j) where $i, j \in N$. If G is a *directed graph* then $(i, j) \neq (j, i)$, but if G is an *undirected graph* then $(i, j) = (j, i)$.

A *subgraph* of $G = (N, A)$ is a graph $G' = (N', A')$ if $N' \subseteq N$ and $A' \subseteq A$. It is a *spanning subgraph* of $G = (N, A)$ if $N' = N$. A *tree* is a connected graph that contains no cycles. A *subtree* is a connected subgraph of a tree. A *spanning tree* of G is a tree that is a spanning subgraph of G and has exactly n-1 arcs. A *minimum spanning tree (MST)* is a spanning tree of minimum cost.

A *cut* is a partition of node set N into two parts, K and $\bar{K} = N - K$. A cut defines the set of arcs that have one endpoint in K and the other in $\bar{K}$.

The *degree* of node i, deg_i, is the total number of incident arcs to node i. The *cost* of arc (i, j) is denoted c_{ij}. In some of the problems that we consider in this thesis, we associate each abstract node i with a location $x^i = \left(x_1^i, x_2^i, \ldots x_d^i\right)$ in the d-dimensional Euclidean space $\Re^d$. In such cases, the cost of arc (i, j) is simply the Euclidean distance, $c_{ij} = \sqrt{\sum_{k=1}^{d}\left(x_k^i - x_k^j\right)^2}$.

A. CLASSICAL RANDOM GRAPH MODELS

The modern treatment of networks was forged by Paul Erdős and Alfréd Rényi (1959), who examined a class of random graphs denoted $G(n, p)$. In this construction, there are n nodes and each node has a probability p of connecting to any other node in the graph. By altering the parameter *p*, the measurable properties of the connectedness of random graphs change quite suddenly (see Bollobás, 1985, for an in depth review). For small values of *p*, the graph demonstrates low connectivity with several isolated nodes. Interestingly though, as *p* approaches *1/n*, a majority of nodes form a cluster and the graph becomes almost completely connected. For values $p \approx 1$, the graph becomes highly connected with several cycles. This phenomenon is known as the "emergence of the giant component." Figure 1 illustrates this phenomenon.

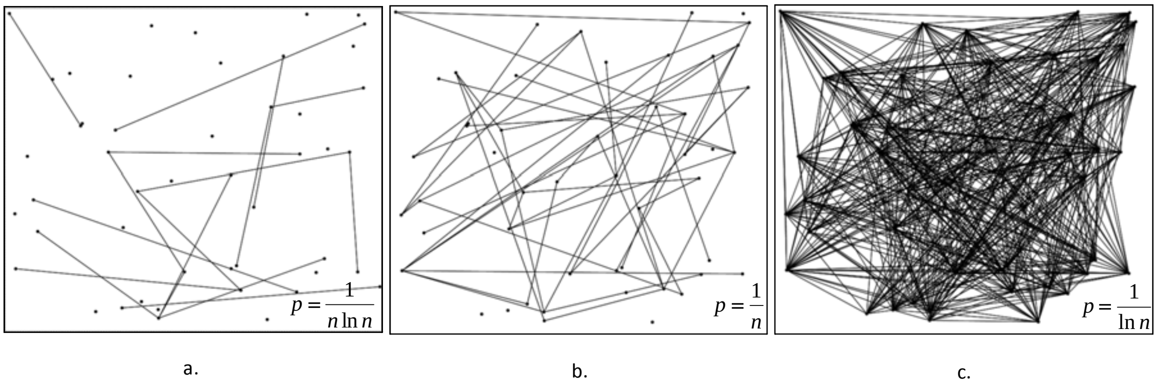


Figure 1. Erdős-Rényi random graphs.

Erdős-Rényi random graphs with varying probabilities for a network with *n* = 50. A small $p = 1/(n \cdot \ln n)$, demonstrates a largely disconnected graph (a). A $p = 1/n$ results in an almost connected graph (b) and a large $p = 1/\ln n$ results in a nearly complete graph (c).

Another important property is the distribution of the node degrees. The degree of node *i*, deg_i, follows a binomial (*n*-1, *p*) distribution. For large values of *n*, this distribution can be approximated with the Poisson distribution (Albert and Barabási, 2002).

B. RANDOM GEOMETRIC GRAPH MODELS

Another approach to generating random graphs builds on the notion of a *geometric graph*. Given a set of nodes *N* indexed *i* = 1,2,…,*n*, having locations $\left\{x^1, x^2, \dots x^n\right\}$, and a positive parameter *r*, the geometric graph $G\left(N, r\right)$ is the undirected graph induced by all arcs (*i*, *j*) having distance $c_{ij} \le r$. When the locations $\left\{x^1, x^2, \dots x^n\right\}$ are the result of an independent and identically distributed (IID) random process, the resulting graph is called a *random geometric graph*.

Most of the theoretic results for random geometric graphs are cumbersome, especially in higher dimensions (see Penrose, 2003, for an in-depth treatment). Unlike classical Erdős-Rényi graphs in which the presence of arcs is independent, the role of proximity in random geometric random graphs makes the appearance of (nearby) arcs dependent. However, these graphs share remarkably similar behavior in the emergence of the giant component (see Goel, Rai, and Krishnamachari, 2003, and references therein).

Random geometric graphs are often used in classification problems in statistics. For example, suppose that individuals have *d* characteristics and each can be represented by a continuous variable (which may not be true in practice). Using an appropriately defined measure of distance in this *d*-dimensional space, one can classify two individuals as being “similar” if their distance is less than some constant parameter *r*. This approach makes it possible to identify clusters of similar individuals, which can be useful in many practical applications.

Figure 2 illustrates the features of a random geometric graph.

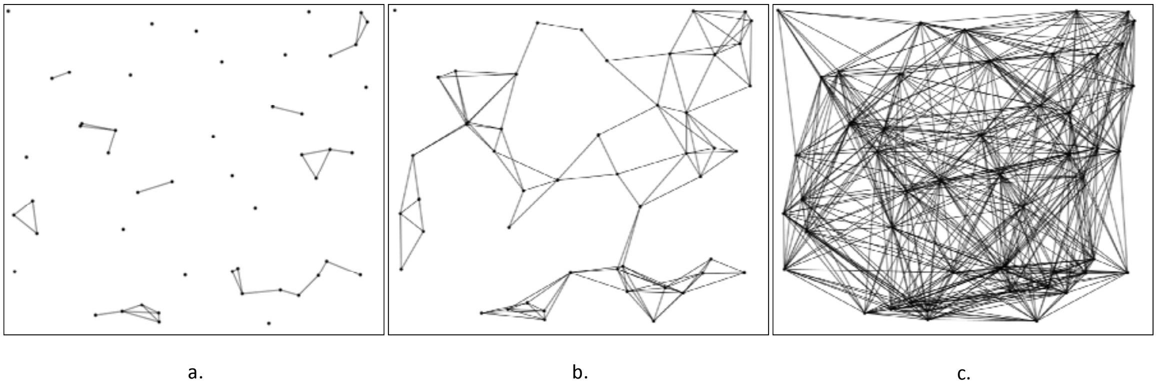


Figure 2. Random geometric graphs.

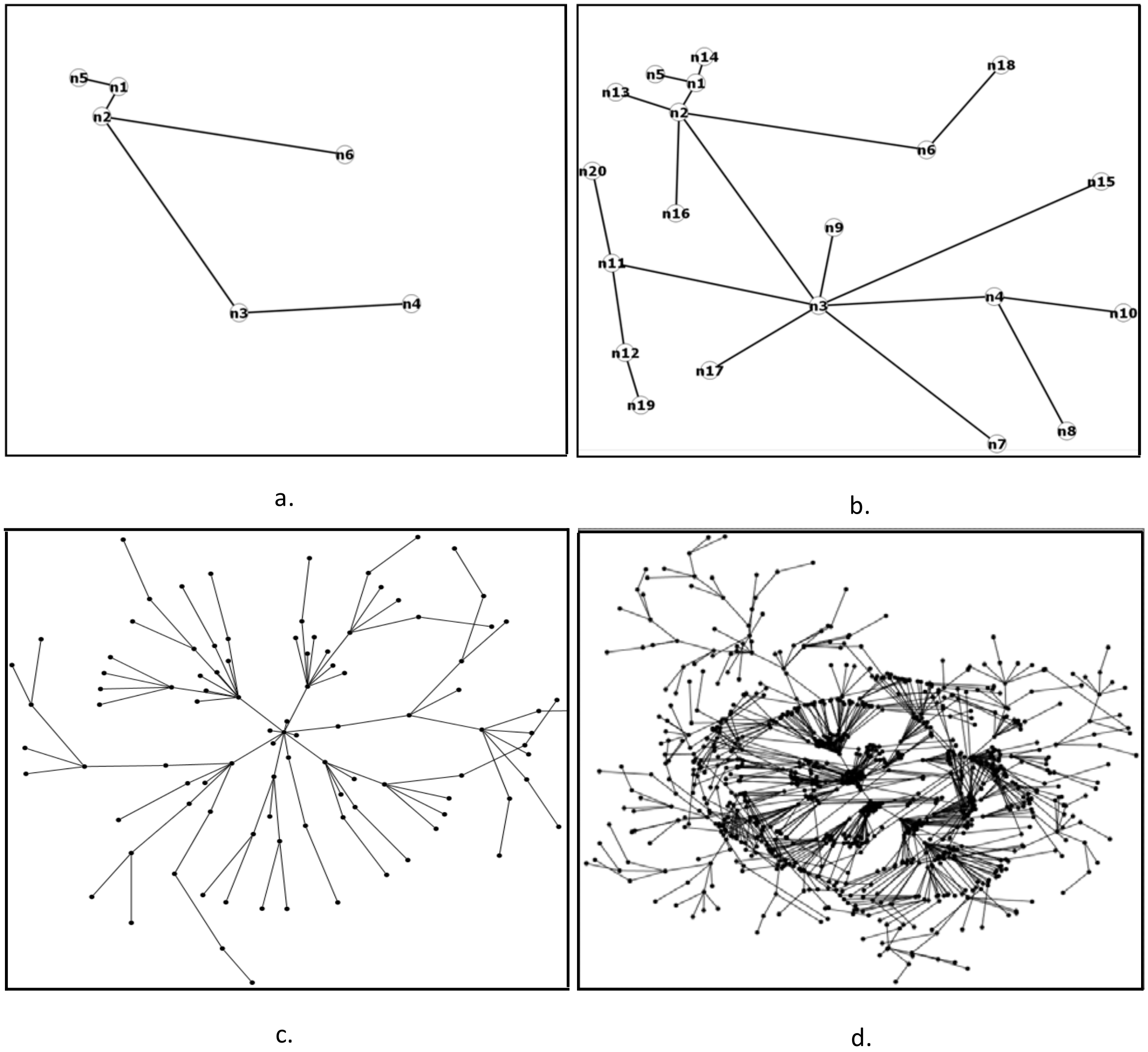
Random geometric graphs of *n* = 50 nodes resulting from varying values of the parameter *r*. A sparse graph (a) results when *r* = 0.1. The emergence of the giant component when *r* = 0.2 (b). A highly connected graph (c) for *r* = 0.5.

C. PREFERENTIAL ATTACHMENT MODELS

Unlike the graphs produced in the Erdős-Rényi model, the degree distribution of many real world networks does not follow a Poisson distribution. Albert and Barabási (2002) observe that many networks have a skewed distribution, in which the majority of nodes have small node degrees while very few nodes have high degrees. The connectivity of these networks can be characterized by a *power-law* distribution, in which the probability that a node has a degree distribution *k* is $P(k) \approx k^{-\gamma}$, where typically $2 \leq \gamma \leq 3$.

Research in a multitude of disciplines has demonstrated power law distributions within networks. Price (1965) demonstrates that the network of bibliographic citations in scientific publications produces a heavy tailed distribution. West (1999) argues that several characteristics of biological systems, such as metabolic rate, sizes and time scales can be modeled with a power-law for several species. Faloutsos, Faloutsos and Faloutsos (1999) argue that power-laws could be used to predict characteristics of the Internet topology. In finance and economics, Gabaix (2009) provides a good summary of power law distributions exhibited in a variety of areas such as firm size, city size, and the distribution of income and wealth.

Since the random graph model does not produce the power-law distribution of node degrees as observed in real world networks, Barabási and Albert (1999) present a different model based on *preferential attachment*. This model differs from the random graph model by accounting for network growth, and it assumes that newly added nodes are more likely to attach to nodes with high connectivity. The probability p that the new node will attach to node i depends on the connectivity of node i, where connectivity is the proportion of node i's degree to the sum of the degrees of all other nodes, such that $p = deg_i / \Sigma_j deg_j$. This model of network formation produces a *scale-free network*, a graph whose resulting node degree distribution follows a power-law. For some systems, the scale-free network produced by the model is more similar in its connectivity than a graph generated from the random graph model. Figure 3 shows the formation of a 1000 node model via preferential attachment, while Figure 4 displays the resulting power-law of the node degree distribution.



a.

b.

c.

d.

Figure 3. Graph formed from preferential attachment model.

Growth of a graph with *n* = 1000 nodes using the preferential attachment model described by Barabási and Albert (1999). Nodes initially are added to the network (a). Initial hubs begin to form (b). Larger hubs are easily identifiable for networks with 100 nodes (c) and 1,000 nodes (d).

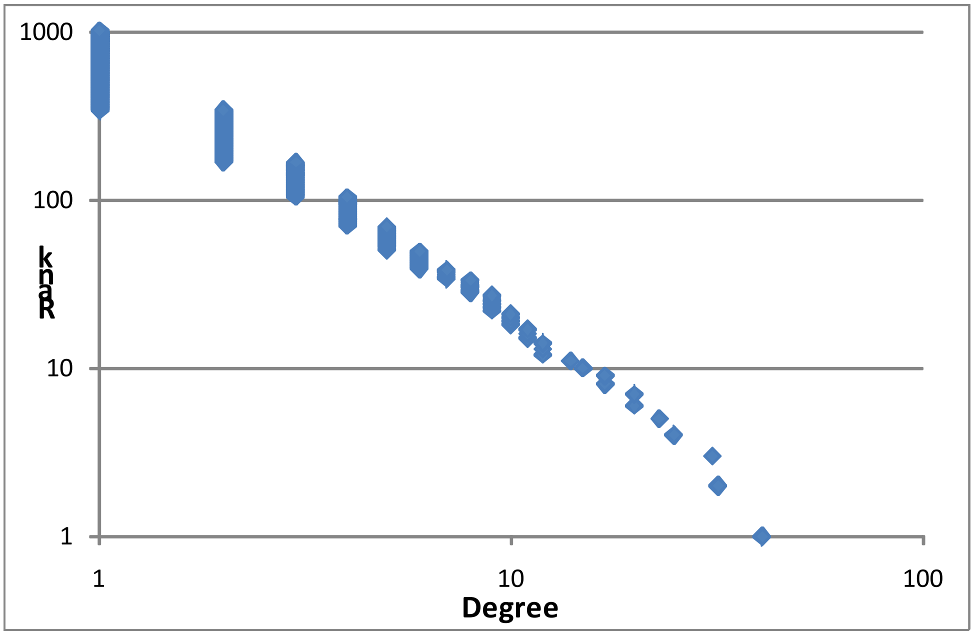


Figure 4. Degree distribution of network with n = 1000 nodes from Figure 3d.

Nodes are ranked based on their node degree from 1 to 1000, with the node with the highest degree ranked as 1, then plotted versus their degree. Because of the large number of nodes with small node degrees, there is a lack of resolution in this region of the plot.

D. OPTIMIZATION-BASED MODELS

Doyle and Carlson (1999) propose a different mechanism, called *highly optimized tolerance* (HOT) that produces power-law distributions. They suggest that complex networks are optimized for robust performance and that the observed power law distributions are a result of the tradeoffs that must be made due to system constraints. Key features of their HOT model include "(1) high efficiency, performance and robustness to designed-for uncertainties; (2) hypersensitivity to design flaws and unanticipated perturbations; (3) nongeneric, specialized, structured configurations; and (4) power laws" (Doyle and Carlson, 1999, p. 1413).

Fabrikant, Koutsoupias and Papadimitriou (2002) suggest a simple model, which we will refer to as the *FKP model*, for network formation that is based on the "tradeoff" concept present in the HOT model. Like the Barabási-Albert model, they grow a network one node at a time, but they also give each node a location in the unit square. When deciding to which node in the network the new node should attach, they propose two logical considerations. First, they assume the node would want to minimize its connection "cost" (represented by the Euclidian distance between itself and the node it attaches to). The second is that the node would desire to connect to one that is more centrally located. These objectives can be weighted in order to alter the relative importance between the two. Specifically in their model, node *i* will attach to node *j* in order to satisfy the objective:

$$\min_{j:\, j<i} \alpha \cdot c_{ij} + h_j \qquad (1)$$

where c_{ij} is the Euclidean distance between nodes *i* and *j* and h_j is a measure of *centrality* for node *j*. The weighting factor, $\alpha \geq 0$, is usually defined as a function of the final number of nodes *n*. The centrality *h* can be defined in several ways, such as the average number of hops to all other nodes, the average Euclidian distance to all other nodes or the distance to some predefined central node (Fabrikant et al., 2002).

Fabrikant et al. (2002) show that by varying the value of α, graphs with very different properties result. They prove that when centrality is measured as the number of hops to a defined node, n0, then for α < 1/$\sqrt{2}$, distance is relatively insignificant compared to centrality, and the resultant network is a star with the center at n0.. As α approaches $\sqrt{n}$, there is a closer trade-off between distance and centrality, and the node degree distribution can be represented by a power law. Once α exceeds $\sqrt{n}$, distance becomes the overriding factor, and a form of a Euclidean spanning tree results. Figure 5 displays 1000 node networks formed with α = .75, 5 and 32 ($\approx \sqrt{1000}$). Figure 6 demonstrates the power law that results when α is approximately $\sqrt{n}$.

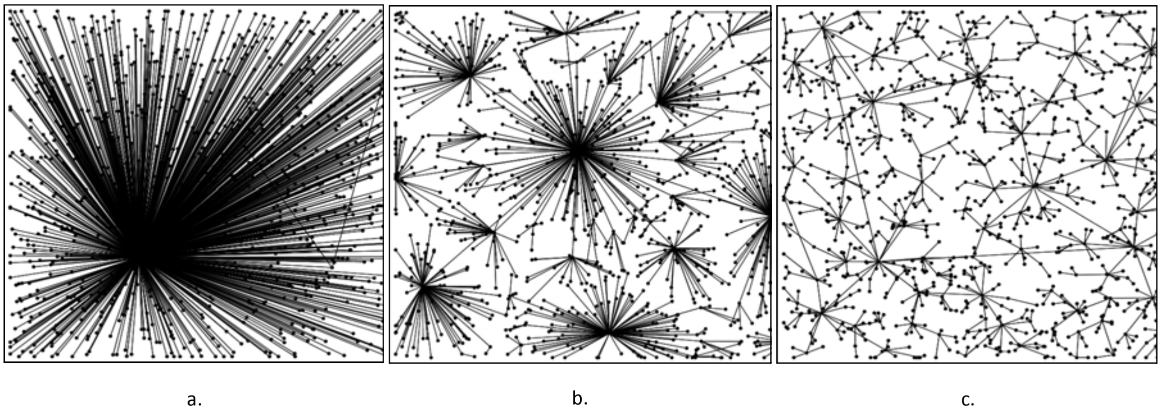


Figure 5. Some realizations of FKP networks with n = 1000.

The star results (a) with a small α = 0.75. Large clusters are evident (b) when α = 5. A network whose node degree distribution can be represented by a power-law (c) results when $\alpha = 32$ ($\approx \sqrt{n}$).

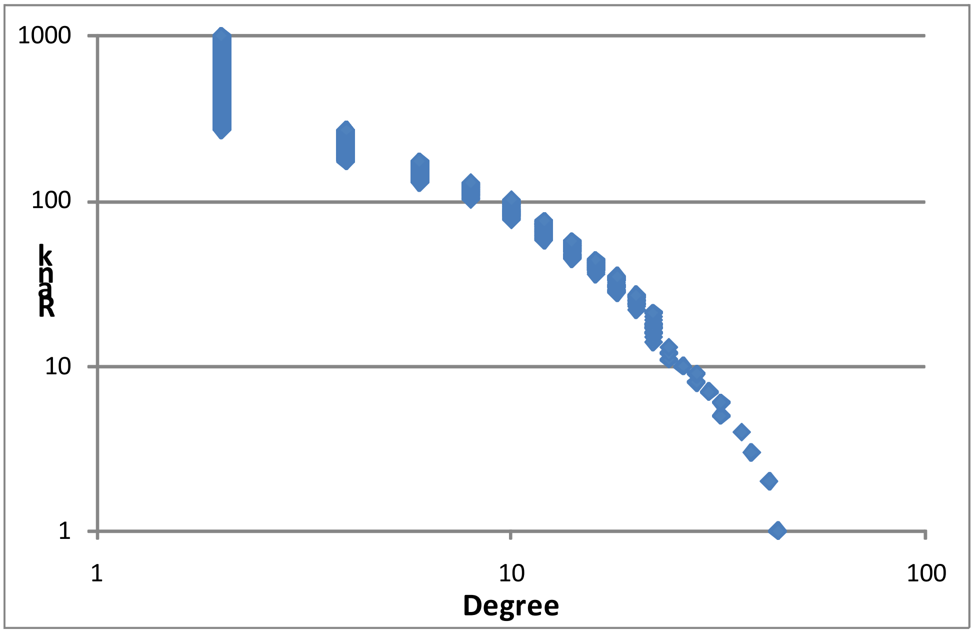


Figure 6. Degree distribution of an FKP network.

Network has $n = 1000$ nodes and $\alpha = 32$ ($\approx \sqrt{n}$). Nodes are ranked based on their node degree from 1 to 1000, with the node with the highest degree ranked as 1, then plotted versus their degree. Because of the large number of nodes with small node degrees, there is a lack of resolution in this region of the plot

The FKP model introduces a novel idea for network formation. Unlike the Erdős-Rényi graphs that are entirely based on a random selection of arcs, this model suggests a highly organized, locally optimized model. The model also demonstrates a power-law in the node degree distribution as seen in the Barabási-Albert model, although Berger, Bollobás, Borgs, Chayes, and Riordan (2003) argue that the resulting distribution is not a strict power-law, but has an exponential cutoff.

E. GAME THEORETIC MODELS

Another approach to network formation uses game theory. Fabrikant, Luthra, Maneva, and Papadimitriou (2003) propose a network formation game to explore how an undirected network created from selfish-acting nodes would affect the network performance as a whole.

The game is as follows: There are *n* players, each representing a node in the network. The entire set of players is $N,$ with $|N| = n$. Each player $i \in 1, 2, ..., n$ chooses a strategy set $s_i = \{s_{i1}, s_{i2}, ..., s_{ij}, ..., s_{in}\}$, which defines the network edges to build from *i* to other nodes $j \in 1, 2, ..., n$. The set $s = \{s_1, s_2, ..., s_n\}$ denotes the collective strategy of all players.

Let *A(s)* be the set of arcs resulting from strategy *s*. Therefore, $A(s) = \{(i, j) : i \neq j,\ s_{ij} = 1 \text{ or } \mathrm{s}_{ji} = 1\}$ and $G(s) = (N, A(s))$ is the undirected graph that results from strategy *s*. Once a strategy is chosen, each player $i \in 1, 2, ..., n$ incurs a cost $c_i(s) = \alpha \cdot |s_i| + \sum_{j \in N} d_{(i,j)}(G(s))$ where *α* is the fixed cost of forming a single connection between two players, and $d_{(i,j)}(G(s))$ is the distance, measured in hop count, between nodes *i* and *j* in the resulting graph $G(s)$. If no path exists between *i* and *j*, then $d_{(i,j)}(G(s)) = \infty$. This is called the *Unilateral Connection Game* (UCG) because each node is able to use an arc, regardless of who paid it.

An extension of this game is the *Bilateral Connection Game* (BCG) described by Corbo and Parks (2005). The major difference from the UCG is that in the BCG an arc only forms if both nodes' strategies contain that arc. So in this game $A(s) = \left\{(i, j): i \neq j,\ s_{ij} = 1 \text{ and } \mathrm{s}_{ji} = 1\right\}$ and any connection cost is shared equally between the two nodes.

In both cases, the *social cost* of the network is compared to that of a *Nash equilibrium*. The Nash equilibrium is a strategy *s* that satisfies

$$c_i(s) = c_i(s_i, s_{N\backslash i}) \leq c_i(s_i', s_{N\backslash i}) \quad \forall i \in N,\ s_i \in S_i\,.$$

In other words, at the Nash equilibrium, no node has incentive to change its strategy. The *social cost* of the network is defined as:

$$C(G(s)) = \sum_{i \in N} c_i(s) = \begin{cases} \alpha \cdot |A(s)| + \sum\limits_{i,j \in N} d_{(i,j)}(G(s)) & \text{for the UCG} \\ 2\alpha \cdot |A(s)| + \sum\limits_{i,j \in N} d_{(i,j)}(G(s)) & \text{for the BCG.} \end{cases}$$

The term price of anarchy is the ratio of the social costs of the worst case Nash equilibrium and the social optimum (Koutsoupias and Papadimitriou, 1999; Roughgarden, 2005) and is used to measure the lack of coordination when the nodes act selfishly.

Fabrikant et al. (2003) show that the results of the UCG vary based on the value of the parameter *α*. When *α* < 1, the social optimum is a complete graph, and this is the only Nash equilibrium. When 1 ≤ *α* < 2, the complete graph still results in a Nash equilibrium, but it is no longer unique. The worst Nash equilibrium is the star, leading to a price of anarchy of $\dfrac{C(star)}{C(complete\ graph)} \leq \dfrac{4}{3}$. When *α* ≥ 2, the social optimum is a star, although there can be worse Nash equilibria.

Corbo and Parkes (2005) build upon the equilibrium concept to define a pairwise Nash equibrium as a strategy s that supports $G(s)$ as a Nash equilibrium and for all $(i, j) \notin A(s)$, if $c_i(s + \Lambda_{(i,j)}) \leq \mathrm{c}_i(\mathrm{s})$ then $c_j(s + \Lambda_{(i,j)}) > c_j(s)$ where $\Lambda_{(i,j)}$ represents a strategy consisting of only the arc (i,j). They also define a network $G(s)$ as being pairwise stable if for all $(i, j) \in A(s)$, $c_i(s - \Lambda_{(i,j)}) \geq c_i(s)$ while for all $(i, j) \notin A(s)$, if

$c_i(s+\Lambda_{(i,j)}) \leq c_i(s)$ then $c_j(s+\Lambda_{(i,j)}) > c_j(s)$. Similarly to the UCG, they prove that for $\alpha < 1$, the complete graph is the only efficient and pairwise stable graph. For $\alpha > 1$, the star network is the only efficient graph, but although stable, it is not unique. They also propose that the price of anarchy for the BCG is worse than that of the UCG (Corbo and Parkes, 2007).

Fabrikant et al. and Corbo and Parkes focus on different properties of the networks formed from the UCG and BCG than the previously reviewed models. Similar to the HOT model, the connection cost is associated with Euclidian distance and the number of arcs in the network and by tuning the weighting factor *α*, networks ranging from the complete graph to a star can be produced. However, unlike the previous models, they concentrate on quantifying the cost associated with selfish behavior to compare it to the social optimum. Also, the BCG introduces a unique feature of restricting the arcs in network to those formed through the consent of the two nodes. Both models provide an interesting way of looking at network formation.

F. DISCUSSION

The key insight of Fabrikant et al. (2002) is that the power laws observed in the structure of many man-made and natural systems can result from design tradeoffs that can be captured in simple optimization models. Their model was inspired by tensions perceived in the Internet—a desire to minimize the cost of connecting while also wanting to have low delay (i.e., be "central") when communicating. But their model reflects a local, myopic decision process. It is unclear how, if at all, this local process relates to the global behavior of the network.

The price of anarchy in the study of network formation games addresses explicitly the difference between the social optimum for some system (as achieved, for example, by a central decision maker) with the aggregate outcome of local agents. Fabrikant et al. (2002) focus on the global connectivity properties (e.g., degree distributions) of the graph, but is there an interpretation for a system-wide objective?

In the case of large α the objective (1) emphasizes only the local connection cost, and it is possible to interpret the collective behavior as trying to minimize the distance of the resulting tree, albeit in a heuristic manner.   With this in mind, we now consider the classic minimum spanning tree problem.

THIS PAGE INTENTIONALLY LEFT BLANK

III. THE MINIMUM SPANNING TREE PROBLEM

A. INTRODUCTION TO MINIMUM SPANNING TREE

A classic problem in the study of networks is the minimum spanning tree (MST) problem with the first algorithm for solving the MST published by Otakar Borůvka in 1926 (see Graham and Hell, 1985, for history of MST). Minimum spanning trees have several practical applications such as determining the minimum amount of wire to connect several electrical components or calculating the minimum amount of piping required to connect houses in a neighborhood to a water system.

The MST problem for a network *G(N,A)* can be treated as an optimization problem and easily be formulated as an integer linear program. The formulation follows, where $H \subseteq N$ and the arc set for *H*, $A(H) \subseteq A$.

1. Indices

$i \in N$ node (i = 1,2,…,*n)* (alias *j*)

$(i, j) \in A$ undirected arc between node *i* and node *j*

2. Data

c_{ij} cost of arc (*i*, *j*)

3. Decision Variable

Z_{ij} binary variable indicating if arc (*i*, *j*) is in tree

4. Formulation

$$\min_{Z} \sum_{(i,j)\in A} c_{ij} Z_{ij} \qquad (3.1)$$

s.t.

$$\sum_{(i,j)\in A} Z_{ij} = n-1 \qquad (3.2)$$

$$\sum_{(i,j)\in A(H)} Z_{ij} \le |H|-1 \qquad \forall \text{ sets } H \subseteq N \qquad (3.3)$$

$$Z_{ij} \in \{0,1\} \qquad (3.4)$$

The objective function (3.1) minimizes the costs of the arcs chosen. Equation 3.2 is a cardinality constraint that ensures that only *n*-1 arcs are selected, while Equation 3.3 ensures that there are no resulting cycles. Although this problem is simple to formulate, solving it with linear programming is nontrivial for large *n*. The number of sets $H \subseteq N$ grows exponentially with *n*, so the total number of constraints arising from Equation 3.3 becomes exponential, making the problem increasingly difficult to solve as *n* grows.

However, the MST has a special tree structure that results in two necessary and sufficient conditions to prove that a tree is a MST (see Ahuja et al., 1993, pp. 518-519, for a detailed discussion). The first is the *cut optimality condition*, which states:

> A spanning tree *T* is a MST if and only if for every tree arc $(i, j) \in T$, $c_{ij} \leq c_{kl}$ for every arc (k, l) contained in the cut formed by deleting arc (i, j) from *T*.

The second is the *path optimality condition*, which states:

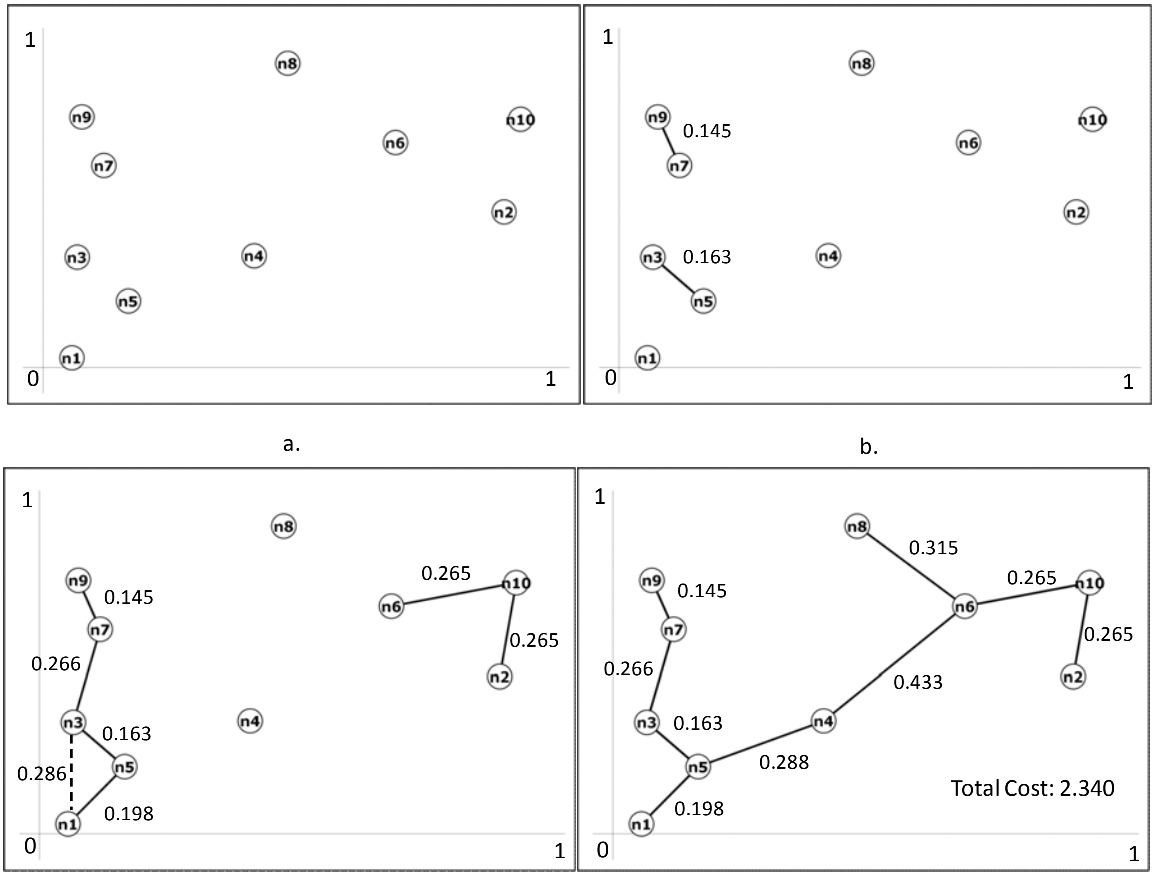
> A spanning tree *T* is a MST if and only if for every nontree arc $(k, l) \in A$ of *G*, $c_{ij} \leq c_{kl}$ for every arc (i, j) contained in the path in *T* connecting nodes *k* and *l*.

Using these two principles for optimality, one can obtain simpler algorithms that solve the MST. *Kruskal's algorithm* and *Prim's algorithm* are two popular methods that use global information of the network to solve for the MST. A novel, decentralized algorithm that utilizes information only known locally to individual nodes has been proposed by Gallager, Humblet and Spira (1983). The following sections of this chapter discuss these algorithms.

B. KRUSKAL'S ALGORITHM

Kruskal's algorithm (1956) directly utilizes the path optimality condition to build the MST one arc at a time. The algorithm does this by maintaining two lists of arcs. The algorithm initializes by sorting all the arcs in increasing order of cost and placing them in a list, called SORTED. A second list, called FINAL, is initially empty. The algorithm proceeds by examining each arc in SORTED and either adding it to the FINAL list or discarding it. The algorithm begins by adding the first arc in SORTED (having the smallest cost) to FINAL. Then for each subsequent arc, the algorithm examines whether adding it to FINAL would create a cycle. If adding the arc would create a cycle, it is discarded. Otherwise, it is added to FINAL. This process continues until there are n-1 arcs in FINAL. This algorithm, as presented, requires $O(m \log n)$ time to sort the arcs and $O(nm)$ time to detect a cycle, although Ahuja et al. (1993) provide a more efficient algorithm that operates in $O(m + n \log n)$ time.

Figure 7 illustrates Kruskal's algorithm on an undirected graph having $n = 10$ nodes, each positioned in the unit square. The cost of the arc between nodes *i* and *j* is the Euclidian distance between the nodes. Here, we assume no restrictions in this example, so each node can connect to any other node. Figure 7a shows the nodes in the initial empty graph. Figure 7b illustrates the first two smallest arcs between nodes *n*7 and *n*9 as well as *n*3 and *n*5. The first arc in SORTED that would create a cycle is the (*n*1,*n*3) arc as shown in Figure 7c. This arc violates the path optimality condition, therefore this arc is discarded. The algorithm examines each arc in turn and adds it if doing so does not create a cycle. Finally, when the number of arcs equals *n*-1 (9 in this example), the algorithm terminates and the MST results (Figure 7d).



a.

b.

c.

d.

Figure 7. Kruskal's algorithm

The initial network has no arcs (a). The smallest arcs are added first (b). The (n1,n3) arc is would create a cycle (c) so it is discarded. The MST results (d).

C. PRIM'S ALGORITHM

Prim's algorithm (1957) is based on the cut optimality condition. It initiates with a cut in, which an arbitrary start node of the network is in subset K, while the remainder of the nodes are in subset $\bar{K}$. The minimum-weight arc from the start node is then added to the list of MST arcs, and the head node of that arc is removed from $\bar{K}$ and placed in K creating a new cut. The minimum-weight edge of all nodes in K that connects to a node in $\bar{K}$ is then added to the MST with its head node moving from K to $\bar{K}$. This method continues to create cuts between the two subsets until all nodes have been placed into K and the resulting MST list will contain n – 1 arcs. Prim's algorithm requires $O(mn)$ time because of the time required to search for the minimum arc in the cut. Ahuja et al. (1993) also present a more efficient data structure that can reduce the time to $O(m + n \log n)$.

Figure 8 illustrates Prim's algorithm on an undirected graph having n = 10 nodes, each with the same coordinates in the unit square as in the previous example. For this algorithm, any node can be chosen to initiate the algorithm, but for this example, node *n*1 initiates the algorithm. Figure 8a shows that *n*1 is in the set *K* whereas the remainder of the nodes are in $\bar{K}$. The hashed, curved line indicates the cut in the graph, and the three smallest cost arcs in the cut are illustrated, although all arcs from *n*1 are technically in the cut. The (*n*1,*n*5) arc has the minimum cost in the cut, so it is added to the MST and the *n*5 node moves from $\bar{K}$ to *K* as demonstrated in Figure 8b. Figure 8b also indicates the four least cost arcs in the new cut and arc (*n*5,*n*3) is the minimum, so it is added (Figure 8c). This process continues to use the cut optimality condition (Figures 8c–e) until all nodes are in the same set and the MST results (Figure 8f).

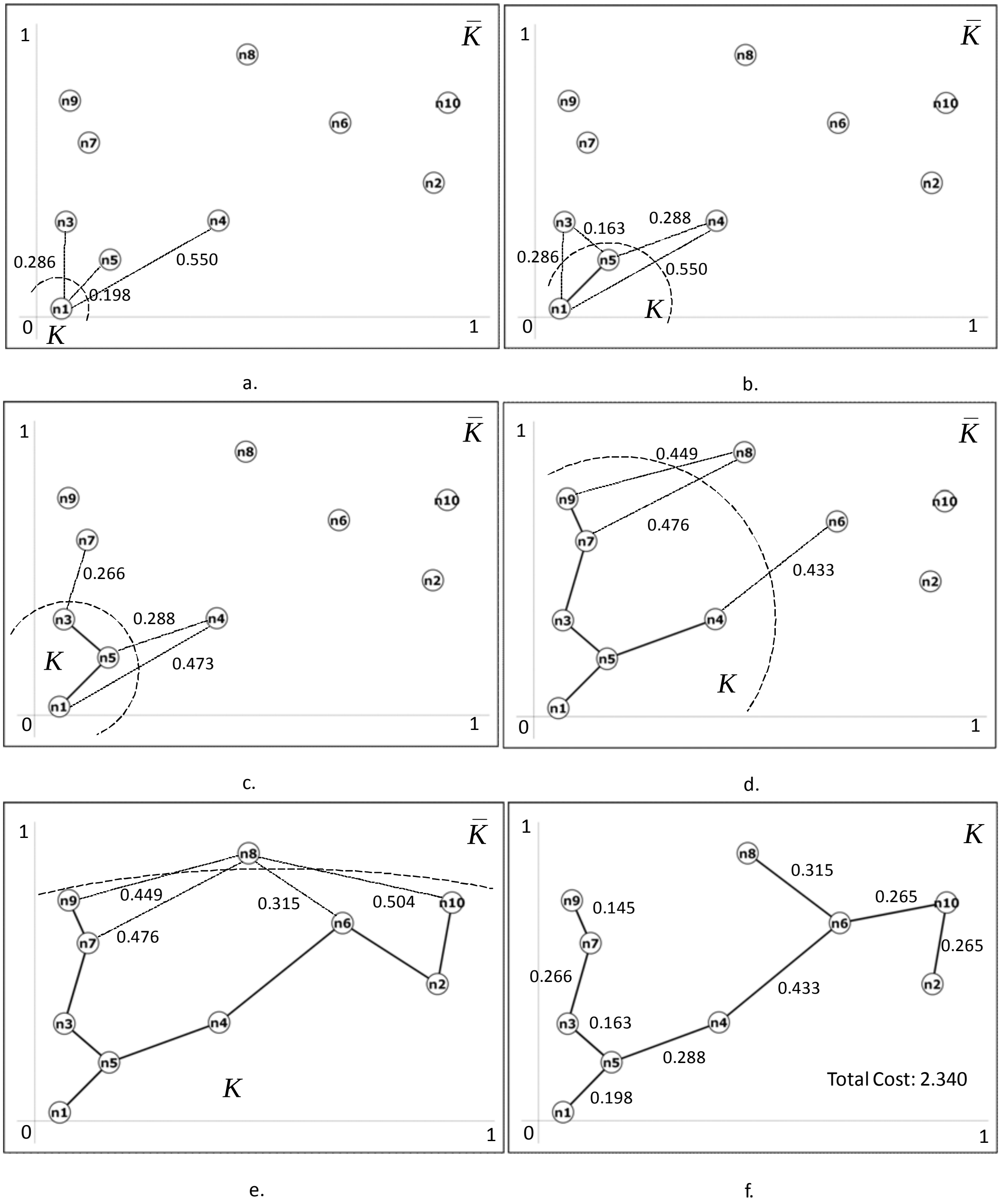


Figure 8. Prim's algorithm

The algorithm begins with one node in subset K and the rest in $\bar{K}$ (a). The minimum cost arc in the cut (denoted by the dashed line) is added to the MST (b) and arcs in the new cut are compared. It proceeds by continually adding the minimum arc in the new cuts (c-e) until the MST is produced (f).

D. DECENTRALIZED ALGORITHM

More recently, focus has turned to decentralized methods for solving the MST problem. Gallager, Humblet and Spira (1983) present a distributed algorithm that also uses the path optimality condition to solve this problem in an undirected network. Their algorithm relies on a node's localized information and its ability to receive, process and send "messages." The complexity of this algorithm is therefore measured by the number of messages that are passed which is at most is at most $5n \log_2 n + 2m$.

Each node maintains an individual queue to store its incoming messages, and it processes them in first-in first-out order. The algorithm works by combining separate graph *fragments* together into a final MST. Initially each node is its own fragment, and then through the message passing process, it combines with other nodes to create new fragments, finally combining into a final fragment containing the MST. By passing messages, each node eventually discovers which of its arcs are in the MST.

Throughout this algorithm, the actions that a node initiates upon receiving a message depends on its *state* and the arc upon which it sends its message depends on the *state* and *weight* of its arcs. A node has one of three states. The *Sleep* state is the initial state for all nodes, the *Find* state is when the node is trying to find its fragment's minimum-weight edge and the *Found* state is for all other instances.

Other information that a node maintains is its fragment *identification*, which is the value of the minimum cost arc joining two fragments, and its current *level*. The level is used to control which fragments can connect. Only a fragment with a lower level can be absorbed by a higher level fragment (not vice-versa) or two fragments of the same level can join. This imposes an ordering on the way that fragments merge.

The algorithm proceeds with nodes passing messages to one another to create fragments along their minimum-weight edges. Each node maintains its own fragment identification, and can send this information in a message. That way, when a node receives a message, it knows if it came from its own fragment or a new fragment. Once fragments have been formed, the nodes continue to pass messages along their arcs to (1) determine the minimum-weight edge of the fragment and (2) see if the receiving node is part of a new fragment. In this manner, the fragments continue to merge until they have obtained the MST.

To keep track of which arcs are involved in the MST, nodes maintain the states of their arcs. Each arc also has one of three states. Initially, all arcs are in the *Basic* state, indicating they have potential to be a part of the MST. If an arc is in the *Branch* state it has been identified part of the MST. And lastly, if it is in the *Rejected* state, then the node has determined that it is not in the MST, but rather an arc connecting two nodes in the same fragment.

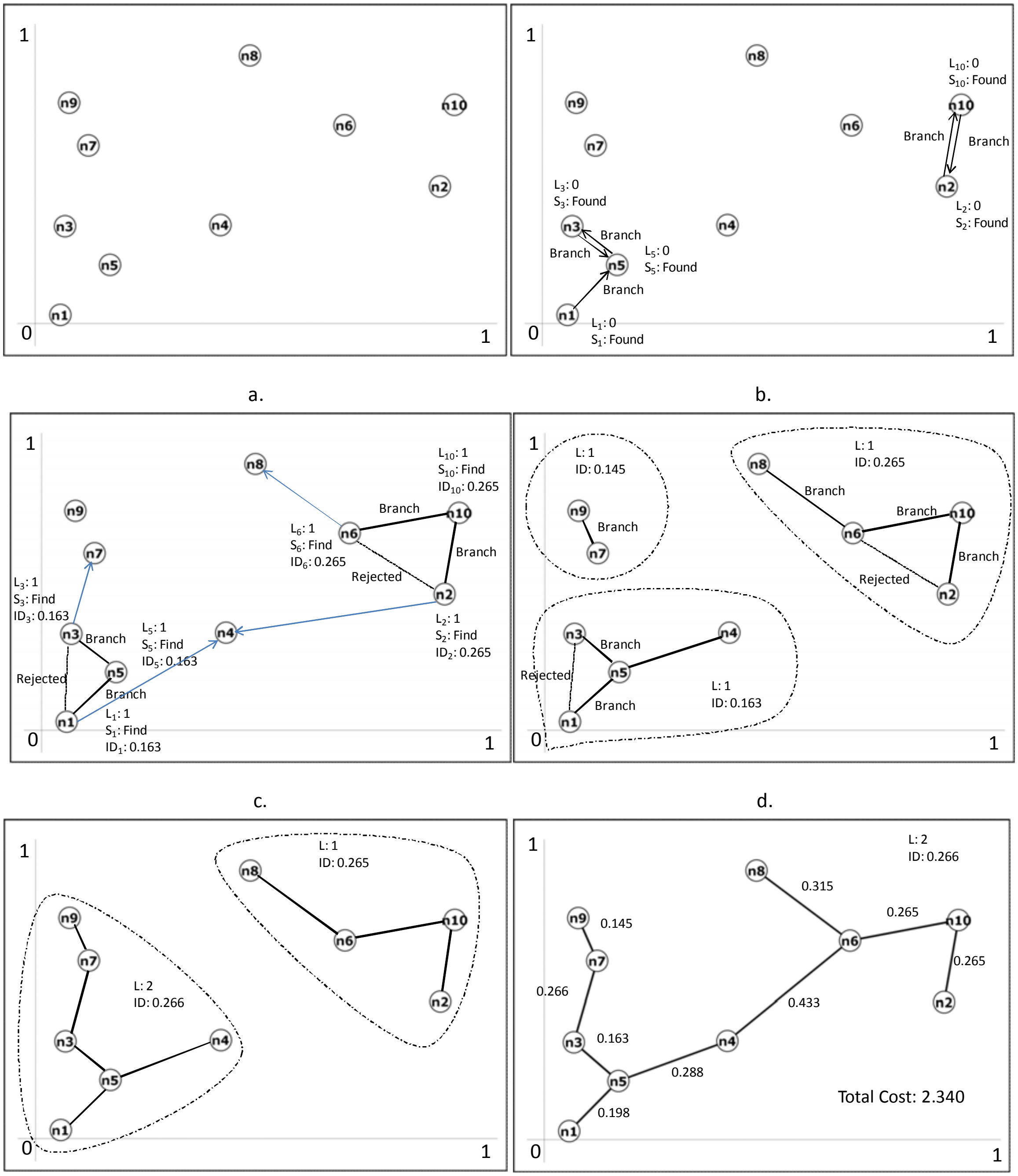
Figure 9 provides an overview (see Gallager et al. 1983, for detailed pseudo-code) of how the decentralized algorithm results in the MST for the same undirected 10 node network used above. Again, it is assumed that each node can attach to every other node in the network, with the weight of the arc between nodes *i* and *j* equal to the Euclidian distance between those nodes. Although this is an undirected network, each node maintains the state of its arcs, so directional arcs will be used to illustrate the arc state maintained by the tail node. It is possible for node *i* and node *j* to have different states for arc (i, j), but the discrepancy is resolved via the message passing and does not interfere with the algorithm.

Figure 9a shows the initialization of the algorithm, in which nodes $n1$ and $n10$ undergo a “wakeup” procedure that initializes their level to 0 and changes their state from *Sleep* to *Found*. Like Kruskal’s algorithm, these nodes each identify their minimum weight arc, $(n1, n5)$ and $(n10, n2)$ respectively, and label them as part of the MST by changing its state to *Branch*. Because the heads of these minimum-weight arcs are in the *Sleep* state, they too must undergo the “wakeup” procedure and find their minimum-weight edges. In this example, nodes $n5$ and $n2$ initialize to level 0 and their minimum-weight edges are $(n5, n3)$ and $(n2, n10)$, respectively. This process continues until, through the message passing process, two nodes find that their minimum-weight edges are one in the same. Because the nodes are at the same level, they combine to form a fragment at a higher level, and the fragment identification is the cost of the minimum-weight edge. Figure 9b illustrates the node and arc states at this point.

Through the message passing, nodes *n*2 and *n*10 recognize that they are in the same fragment as the MST and because they are the same level, they will each set their fragment identification as 0.265 (the weight of the connecting arc) and increase their level to 1 as well as changing their state to *Find*. Nodes *n*3 and *n*5 undergo a similar process, increasing their level to 1 and labeling their fragment identification as 0.163. Node *n5* will also act upon a message received by *n*1. Because *n*1 is still at level 0, it will absorb into the *n*3-*n*5 (0.163) fragment assuming the fragments level and identification as well as changing its state to *Find*.

When the nodes are in the *Find* state, they are actively searching for other nodes to add to their own fragment. They do this by passing a "Test" message along its minimum-weight arc in the *Basic* state. The receiving node compares its fragment identification to that of the sending node. If they are the same, the arc between them is placed in a *Rejected* State for both nodes, and the sending node sends a "Test" message on the next best minimum-weight edge. Figure 9c shows the partial graph formation. The bold arcs are those that both head and tail nodes recognize their state as *Branch* and the gray directional arcs represent the arcs a "Test" message is sent across. Nodes *n*6 and *n*3 both tested nodes *n*2 and *n*1 respectively. Since each of these nodes were in the others fragment, the arc was rejected (the dotted arc) and a new "Test" message is sent via the next best arcs in the *Basic* state, nodes *n*8 and *n*7.

The process continues, resulting in three primary fragments for this example (Figure 9d). The message passing continues between nodes within the same fragment, to identify the minimum-weight arc to connect to a node in a different fragment. For simplicity, we will focus on the message from *n*3 to *n*7. The nodes have different fragment identifications and are at the same level, so they will combine the fragments along arc $(n3, n7)$, increasing their level to 2 and assuming the new fragment identity of 0.266, the weight of the adjoining arc. They will then pass messages along their *Branch* arcs to the other nodes in their fragment so they will update their levels and fragment identifications accordingly. Figure 9e illustrates the graph once this has occurred. The number of rejected arcs at this instance are numerous and are not illustrated. Lastly, the remaining two fragments combine and the result is the MST (Figure 9f).



a.

b.

c.

d.

e.

f.

Figure 9. Decentralized algorithm for finding the MST

All nodes in the initial network (a) are in the Sleep state. Nodes *n*1 and *n*10 wake up and label their minimum arc Branch (b). Fragments form and their nodes pass messages to identify the minimum arc to connect them to each other (c-e) until the MST results (f).

E. DISCUSSION

The MST problem is simple to state and solve as a global optimization problem. However, as a local, decentralized, and asynchronous process, it is considerably more complicated. The algorithm of Gallager et al. (1983) shows that individual nodes making local decisions can solve this problem correctly and efficiently. But, is there a way to interpret this process as a local optimization problem? In the next chapter, we consider some numerical experiments to explore this possibility.

IV. EXPERIMENTS

In the previous chapters, we examined two different classes of network formation problems. In Chapter II, we reviewed a progression of network models that were based on local "decisions" (either random or according to some local optimization problem). The research emphasis for these models has been to understand the global network properties that result from these local decisions. In Chapter III, we considered a specific network design problem, the MST, and reviewed both global and local techniques for solving it. In this chapter, we attempt to reconcile these two perspectives by considering two basic issues.

A. THE ROLE OF PRECEDENCE IN LOCAL NETWORK FORMATION

In many of the network models considered here, the network is not constructed all at once, but rather grows incrementally through the addition of nodes and arcs. A basic question of interest is, *What is the role of precedence in these network formation models?* We consider each of these network models in turn.

Random graph models. The Erdős-Rényi random graph model has one parameter, p, the probability a node connects to any other node in the graph. The random geometric graph model (with node locations generated from an IID random process) also has one parameter r, such that arcs form between nodes with distances $c_{ij} \leq r$. These parameters are not affected by the order that nodes are introduced, and actually, nodes "arrive" at the same time in both of these models. Therefore, in neither of these cases does the order of node arrival affect the overall network structure.

Preferential attachment models. In preferential attachment models, nodes do arrive one at a time. The arriving nodes have a higher probability of forming arcs to nodes that already have several connections. In this model, precedence is important in that nodes introduced early on in the network formation are much more likely to acquire connections than those introduced later on. However, the nodes are essentially interchangeable, so if the order of the nodes were rearranged, the identification of the nodes would change, but the statistical properties of the global network would remain the same, including the power-law distribution of the node degrees. So precedence does not affect the overall network characteristics.

FKP construction. The FKP model also has nodes in unique locations, but in this model precedence does play a role. Whereas the random graph and preferential attachment models use probabilities to determine arc placement, this model adds arcs based on which arc minimizes a node's objective function. Since a newly added node can only utilize the nodes previously added to meet its objective function, different networks will result based on the order the nodes are introduced to the network. What is not known is the extent to which precedence plays a role in the overall network properties.

Network formation games. In the UCG and BCG network formation games, a network is formed by each node picking a strategy consisting of arcs. These models do not utilize network growth, but rather begin with all nodes present in the network. It is unclear how, or if precedence, would affect this type of local network formation.

Minimum spanning tree problems. There is no role of precedence in the MST problems either. With the global algorithms, such as Prim's and Kruskal's, the algorithm can begin with any node and result in a MST. In the decentralized algorithm, the message passing mechanism is not affected by the order in which nodes send messages. Although these methods could produce MSTs with different structures, their costs are all equal. We focus on the total network cost, so the potential for different network structures is unimportant in our numerical experiments.

In order to explore the FKP models in more detail, we conduct two experiments.

1. Reordering of Nodes for Initial Construction of Network

We generate an FKP-style network of n = 100 nodes, each with the local objective function $\min_{j:j<i} \alpha \cdot c_{ij}$ with $\alpha = 1$. The total network cost is the sum of all arc costs. We

compare this result to those obtained from a second network that is generated in the same fashion. Each node in the second network has the identical locations as the nodes in the first network.

We alter, in two ways, the order in which nodes arrive. The first uses the same node sequence, but chooses a different start node. The second method completely randomizes the order of the nodes.

In both cases, the network that forms is different than the initial network and the total network cost is similar to the initial cost. Therefore we conclude that precedence does play a role in the initial network formation. Figure 10 demonstrates the different networks that result from both types of reordering for a 20 node and a 100 node network.

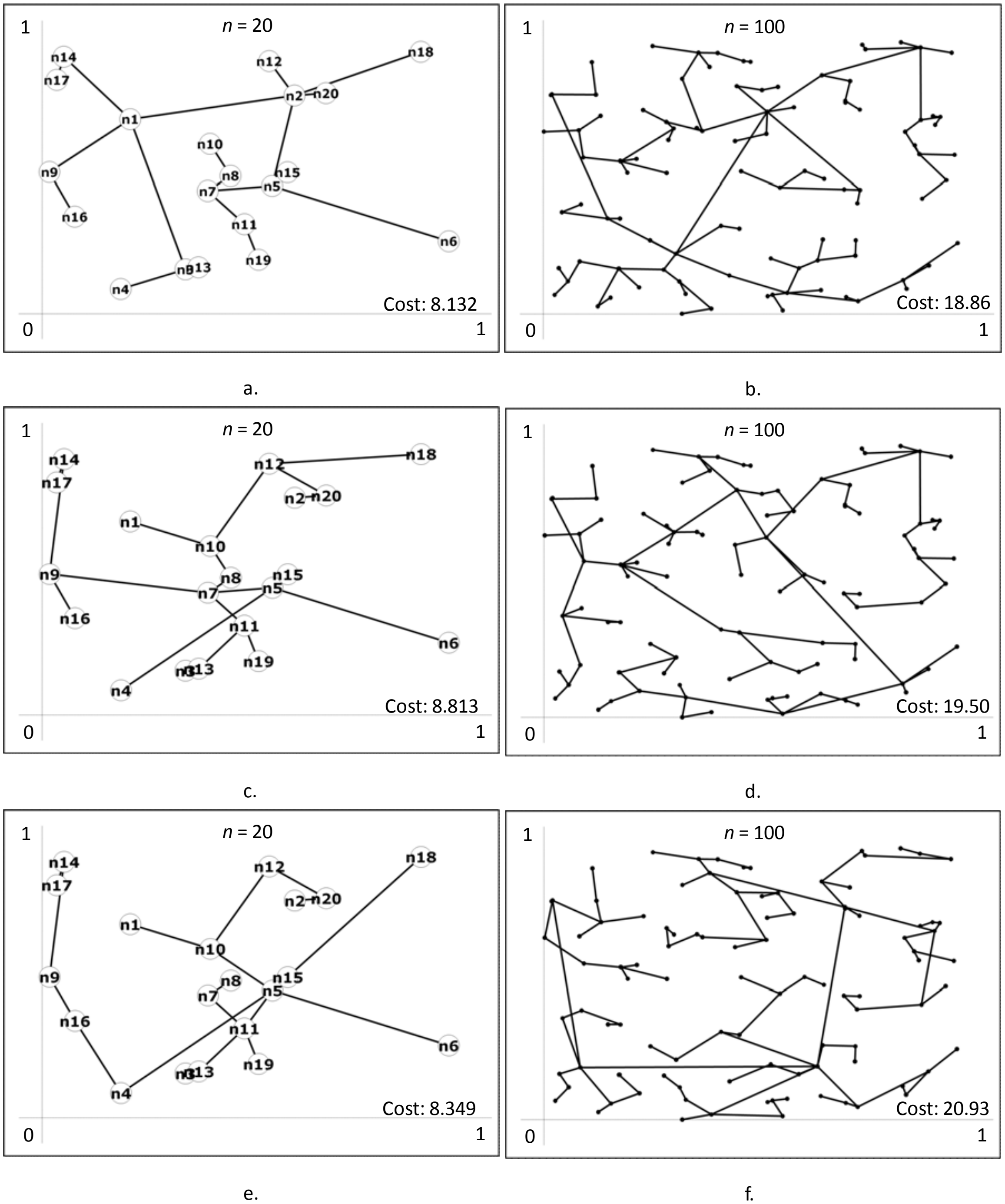


Figure 10. Different FKP-style networks form with reordering the nodes.

Networks of *n* = 20 and *n* = 100 nodes are generated from the FKP-style construction (a,b). Different networks (c,d) result when a start node is randomly selected but the nodes are added in the same sequence. Different networks (e,f) result from randomizing the sequence in which the nodes are added.

2. Rewiring of Nodes to Equilibrium

In the second numerical experiment, we examine the effects of allowing the nodes to change their initial connection. Once all nodes had been introduced to the network, we give them an opportunity to improve its connection cost by selecting a different node to connect with. We term this process *rewiring.* Once none of the nodes in the network can benefit from connecting to a different node we say the network is in *equilibrium*. We then compare the cost of the equilibrium network to those of the MST

We generate networks of n = 100 nodes as described above. Once all nodes are added, we permit the nodes to rewire, subject to two constraints.

The first constraint is that a node is only able to rewire the arc it formed when it joined the network and not any of the arcs from other nodes that attached to it. The second constraint is that the node can only rewire to a node that maintains the connectivity of the entire network.

We alter, in two ways, the order that the nodes rewire. We first give the nodes the opportunity to rewire in the same sequence they arrived in the network. In the second way, we randomly select which node can rewire. Figure 11 illustrates the initial network and the network once it has reached equilibrium.

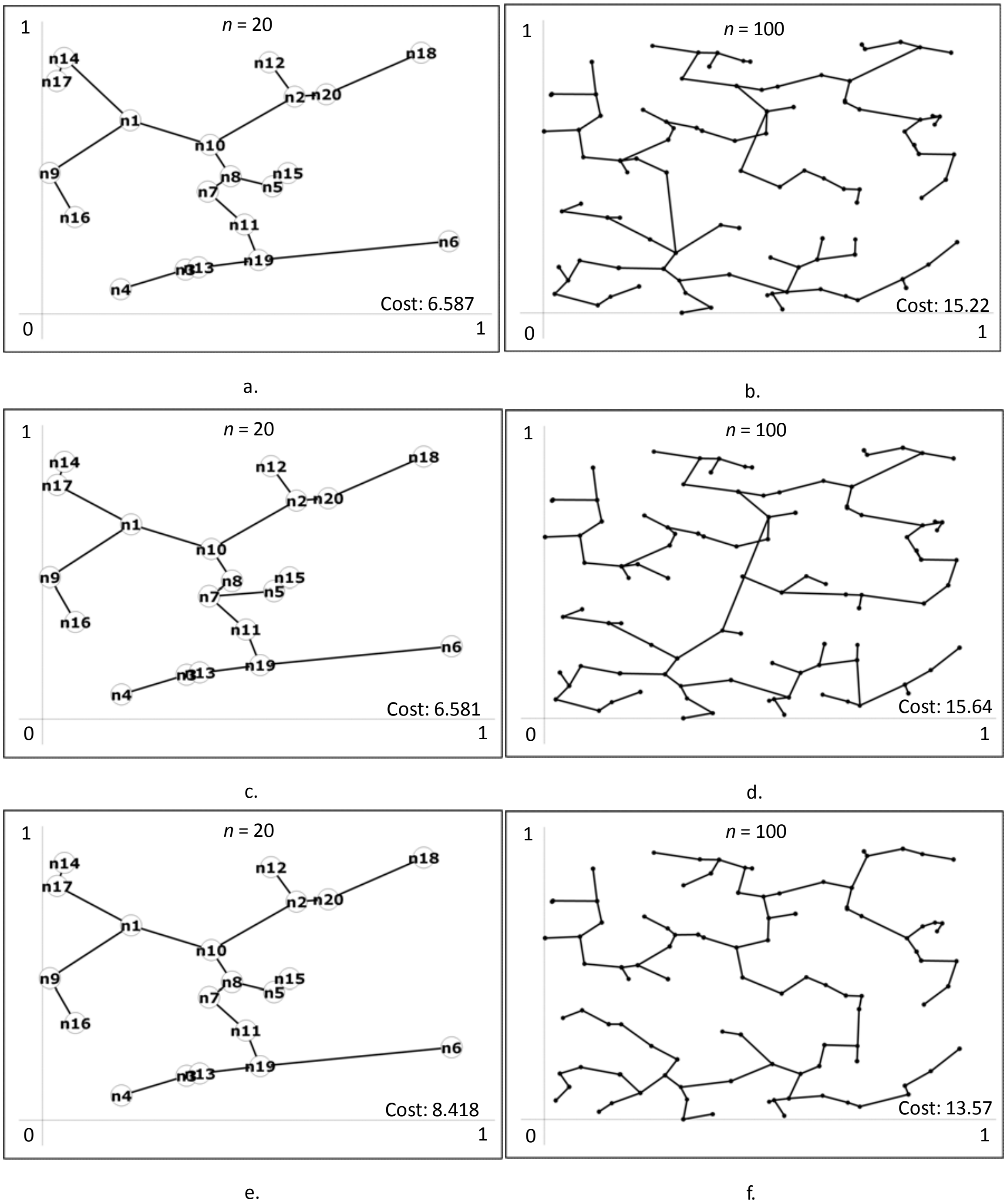


Figure 11.   Networks with FKP-style construction in equilibrium

The 20 and 200 node networks from Figure 10 reach equilibrium by sequentially rewiring their initial arcs (a,b). Different equilibrium networks (c,d) result from a random rewiring process. Neither equilibrium network results in the MST (e,f).

B. COMPARISON OF FKP-STYLE AND MST CONSTRUCTIONS

To examine the price of anarchy, we compare the cost of the equilibrium network to the MST. We repeat the rewiring experiments 10,000 times, generating the initial network, rewiring it to equilibrium, and then determining the MST. Table 1 shows the network costs associated with each type of network and compares them to the cost of the MST. For networks with *n* = 100, the initial network is 46.5% greater than the MST. However, once the network reaches equilibrium, the costs decrease substantially. The network formed from the sequentially ordered method is 14.2% greater than the MST and the network from the randomized method is 15.0% greater than the MST. The difference between the equilibrium network costs from the sequential rewiring method and the randomized rewiring method is not statistically significant. Both the sequential rewiring and the random rewiring methods produced equilibrium networks with the same costs in 26% of the experiments.

Network type	**Average Cost**	**Standard error**	**Ratio to cost of the MST**
Initial FKP-style	19.791	0.874	1.465
Sequential rewiring	15.422	0.609	1.142
Random rewiring	15.533	0.668	1.150
MST	13.506	0.426	1.000

Table 1. Cost comparison of initial network, equilibrium network and MST for *n* =100.

Results based on 10000 trials of networks with *n* = 100. The difference between the costs of the sequentially rewired equilibrium network and the randomly rewired equilibrium network is not statistically significant.

The results indicate that either method of local rewiring can improve the total cost significantly, but the equilibrium value is still considerably worse than the MST. The total network cost is approximately 15% greater than that of the MST, but in none of the 10000 trials is the equilibrium network equal to the MST. This leads to the question: *To what extent, if any, is the method that forms the locally optimized network comparable to the methods that construct the MST?* Since this model does not achieve the MST, we explore what might be preventing it from doing so.

The first issue is precedence in the formation process itself. The experiments demonstrate that the order in which the nodes connect or rewire their connections produces different networks. However, it is conceivable that nodes could be introduced into a network in an order that would result in the model producing the MST. Figure 12a depicts a FKP-style network in equilibrium and Figure 12b is the MST. It is relatively easy to envision an ordering of nodes that would result in this MST. Since it is a tree, imagine starting with the root and moving down the branches. For example, assume node *n*1 is the first node in the network. Because the connection costs for the FKP-style model are based on Euclidean distance, the second node introduced could be *n*10, *n*18 or *n*4 (any node with an arc to *n*1 in the MST). If *n*10 were the second node, then the third node could be *n*4, *n*18 or *n*19, and so on. However, if *n*9 were the second node, it would have to attach to *n*1, the only node in the network, and the final network would not be the MST (although it may appear if the network were allowed to attain equilibrium). As long as the order the nodes are introduced in the network follows the order in the MST, the model would result in the MST.

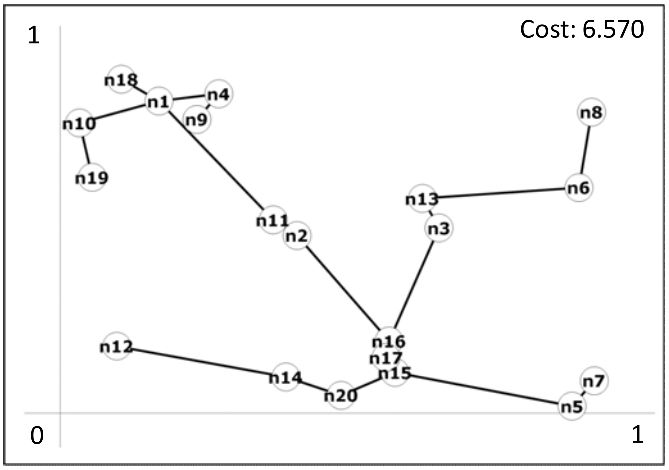
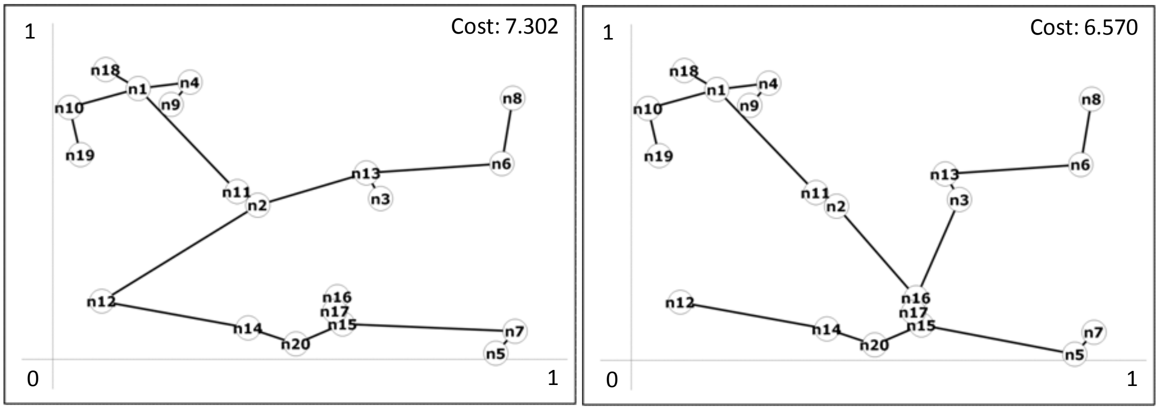


Figure 12. A MST for a network with *n* = 20.

A second reason that the rewiring heuristic may not induce the MST is that it could be too restrictive. Nodes evaluate only one arc at a time, and can only consider rewiring with an arc in the cut produced when it removes the arc it is reevaluating. This constraint is necessary to ensure a connected network. However, there could potentially be a combination of two (or more) arcs that could be rewired simultaneously to achieve the MST.

The other restriction is that a node can also only rewire the arc it formed when joining the network, and not any of the arcs from nodes may have connected to it later on. This might also be preventing the model from developing the MST. This case can be seen in the networks in Figure 13. Figure 13a is the network that is initially formed. In this network, $n2$ has to attach to $n1$, so the $(n2, n1)$ arc is the only one $n2$ is able to reevaluate. Node $n16$ attached to $n15$ so it is only able to evaluate the $(n16, n15)$ arc. When the network is in equilibrium (Figure 13b), $n2$ rewired its arc to $(n2, n11)$ and the $(n16, n15)$ arc becomes the $(n16, n17)$. Both of these changes occurred because the new arcs are shorter. In this method, the $(n2, n16)$ arc, which is in the MST (Figure 12), will never result. However, if $n2$ had the ability to evaluate all its arcs, the $(n2, n12)$ would become the $(n2, n16)$ arc, which is in the MST.



a. b.

Figure 13. FKP-style network with n =20 nodes.

The initially constructed network (a) and its equilibrium network (b).

C. DISCUSSION

This work primarily focuses on incremental network construction based on local, myopic decisions. The major difference between this model and the FKP model is that we removed the tradeoff aspect by not using the centrality term in the local objective function. We simplified the objective function so we can compare the results of the formed networks to the optimal network, the MST, to explore the price of anarchy.

We demonstrate that precedence plays a role in the network formation, producing initial networks with different costs. We also demonstrate that the sequential rewiring process and the random rewiring process substantially improve the network cost, but that there is no statistical difference between the two methods.

The work in this thesis lays the groundwork for more complex numerical experiments and deeper analyses. The heuristic model with rewiring improves the total network cost substantially, but does not achieve the MST. The extent to which the constraints of the model prevent it from obtaining the optimal result is unclear. To explore this, the model could be altered so nodes have the option to rewire all of its incident arcs, not just the one it initially formed.

The model also has room for expansion. We did not use a measure of centrality in the objective equation, but this can be added to the model to explore the effects on the network structure due to trade-offs between centrality and distance. We also primarily focus on forming networks with a tree structure, whereas many real-world networks contain cycles. This model can be altered to allow nodes to form more than one arc when it joins a network.

V. CONCLUSIONS AND FUTURE WORK

Understanding the drivers of complex network formation is nontrivial. In this thesis, we focused on models of network formation with emphasis on both centralized and localized algorithms for solving the minimum spanning tree problem. We developed a local, heuristic model based on a FKP-style construction, which uses rewiring to produce networks in equilibrium. We examined the price of anarchy of these networks due to myopic node behavior. Although the rewired heuristic model does not always produce the MST, we demonstrate that there exist orderings of nodes that can obtain it. This leads to the question: *Is there an interpretation of the local, myopic decision process of the FKP-style construction that lends itself to an equivalent global optimization problem?* If the answer is affirmative, then the local and global methods would both provide the optimal solution and the price of anarchy would be zero. This could have significant implications for the formation of real network systems when global information and central decision processes are not possible.

Is there evidence to suggest that such an interpretation is possible? Here, we appeal to the notion of duality in network optimization problems and note that there is a considerable literature in the use of duality arguments for the development of decentralized algorithms (see Bertsekas and Tsitsiklis, 1997, for an in-depth treatment).

The Internet is an example where duality arguments have recently enhanced our understanding of complex network behavior. The Transmission Control Protocol (TCP) is fundamental to the operation of the Internet. It guarantees end-to-end delivery of data packets by recognizing and retransmitting packets that are lost, and it also controls the rate at which individual computers inject packets into the network. Like most of the protocols used in the Internet, TCP was developed in an ad hoc manner, based on engineering intuition and trial-and-error more than mathematical theory. To researchers in the network science community, the behavior of TCP seemed like a case of self-organization (Veres and Boda, 2000). However, research over the last decade has shown that TCP and its complementary protocol Active Queue Management (or AQM, which runs in routers to manage the size of their limited buffers) work together as a primal-dual algorithm to solve a global resource allocation problem in a decentralized and asynchronous manner (Kelly, Mauloo, and Tan, 1998; Low 2003). This type of analysis is not only bringing greater understanding to the way that the existing Internet works (Srikant, 2004), but it is also helping to influence the design of future network protocols (Chiang, Low, Calderbank and Doyle, 2007).

While there remains considerable work to understand the forces governing complex network behavior, it is clear that optimization is an important tool for exploring the tradeoffs at work in network formation. Identifying the precise mechanisms at work in specific applications, as well as how to improve them, will be a topic of future research.

LIST OF REFERENCES

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network Flows.* Upper Saddle river, NJ: Prentice Hall, Inc.

Albert, R., and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics , 74*, 47–97.

Barabási, A.-L., and Albert, R. (1999). Emergence of scaling in random networks. *Science , 286*, 509–512.

Berger, N., Bollobás, B., Borgs, C., Chayes, J., and Riordan, O. (2003). Degree distribution of teh FKP network model. *Lecture Notes in Computer Science , 2719*, 725–738.

Bertsekas, D. P., and Tsitsiklis, J. N. (1997). *Parallel and Distributed Computation: Numerical Methods.* Belmont, MA: Athena Scientific.

Biggs, N. L., Lloyd, E. K., and Wilson, R. J. (1998). *Graph Theory, 1736–1936.* New York: Oxford University Press, Inc.

Bollobás, B. (1985). *Random Graphs.* London: Academic Press.

Carlson, J. M., and Doyle, J. (1999). Highly optimized tolerance: A mechanism for power laws in designed systems. *Physical Review E , 60* (2), 1412–1427.

Chiang, M., Low, S. H., Calderbank, A. R., and Doyle, J. C. (2007). Layering as optimization decomposition. *Proceedings of IEEE , 95*, 255–312.

Corbo, J., and Parkes, D. (2005). The price of selfish behavior in bilateral network formation. *ACM PODC '05.* Las Vegas, NV.

Denning, P. J. (2005). The social life of innovation. *Communications of the ACM , 49* (4), 15–20.

Erdős, P., and Rényi, A. (1959). On random graphs. *Publicationes Mathematicae , 6*, 290–297.

Fabrikant, A., Koutsoupieas, E., and Papadimitriou, C. H. (2002). Heuristically optimized trade-offs: A new paradigm for power laws in the Internet. *Lecture Notes in Computer Science , 2380*, 110–122.

Fabrikant, A., Luthra, A., Maneva, E., Papadimitriou, C. H., and Shenker, S. (2003). On a network creation game. *ACM PODC '03.* Boston, MA.

Faloutsos, M., Faloutsos, P., and Faloutsos, C. (1999). On power-law relationships of the Internet topology. *ACM SIGCOMM '99.* Cambridge, MA.

Gabaix, X. (2009). Power laws in economics and finances. *Annual Review of Economics , 1*, 255–294.

Gallager, R. G., Humblet, P. A., and Spira, P. M. (1983). A distributed algorithm for minimum-weight spanning trees. *ACM Transactions of Programming Languages and Systems , 5* (1), 66–77.

Goel, A., Rai, S., and Krishnamachari, B. (2005). Monotone properties of random geometric graphs have sharp thresholds. *Annals of Applied Probability , 15* (4), 2535–2552.

Graham, R. L., and Hell, P. (1985). On the history of the minimum spanning tree problem. *Annals of the History of Computing , 7* (1), 43–57.

Kelly, F. P., Maulloo, A., and Tan, D. (1998). Rate control in communication networks: Shadow prices, proportional fairness and stability. *Journal of the Operational Research Society , 49*, 237–252.

Koutsoupias, E., and Papadimitriou, C. H. (1999). Worst-case equilibria. *Lecture Notes in Computer Science , 1563*, 404–413.

Kruskal, J. B. (1956). On the shortest spanning tree of graph and the traveling salesman problem. *Proceedings of the American Mathematical Society , 7*, 48–50.

Low, S. H. (2003). A duality model of TCP and queue management algorithms. *IEEE/ACM Transactions on Networking , 11* (4), 525–536.

National Research Council. (2005). *Network Science.* Washington, D. C.: The National Acadamies Press.

Newman, M. E., Barabási, A.-L., and Watts, D. J. (2006). *The Structure and Dynamics of Networks.* Princeton, NJ: Princeton University Press.

Penrose, M. (2003). *Random Geometric Graphs.* Oxford, UK: Oxford University Press.

Price, D. J. (1965). Network of scientific papers. *Science , 149* (3683), 510–515.

Prim, R. C. (1957). Shortest connection networks and some generalizations. *Bell System Technical Journal , 36*, 1389–1401.

Roughgarden, T. (2005). *Selfish Routing and the Price of Anarchy*. Cambridge: MIT Press.

Srikant, R. (2004). *The Mathematics of Internet Congestion Control.* Boston: Birkhauser.

Veres, A., and Boda, M. (2000). The chaotic nature of TCP congestion control. Proceedings of *IEEE INFOCOM*. Tel-Aviv, Israel.

West, G. B. (1999). The origin of universal scaling laws in biology. *Physica A* , *263*, 104–113.

Willinger, W., and Doyle, J. C. (2004). Robustness and the Internet: Design and evolution. In E. Jen (Ed.), *Robust-Design: A Repertoire of Biological, Ecological, and Engineering Case Studies* (pp. 231–272). New York: Oxford University Press.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. David L. Alderson
   Naval Postgraduate School
   Monterey, California